ESTRUCTURAS DE DATOS

ESTRUCTURAS DE DATOS

- **Lectura destructiva**
  - **Cola**
  - **Pila**
- **Lectura no destructiva**
  - **Lista**
  - **Árbol**

Lectura destructiva

Cada vez que se lee un dato este se destruye

La destrucción del dato puede ser de dos formas.

Física: se borra el dato.

Es un proceso lento.

Lógica: no se borra el dato. Se impide su acceso.

Es un proceso rápido.

Se utilizan para almacenamiento temporal

Pueden ser:

Compactas: los datos están ubicados en posiciones consecutivas en la memoria.

Se utilizan vectores.

Dispersas: la ubicación de los datos no son adyacentes.

Se ubican en distintas posiciones y se accede mediante punteros.

Se utilizan listas.

COLA - BUFFER

El Primer dato que se ingresa es el primero que se lee.

Se utilizan don punteros o índices.

uno de escritura y otra de lectura.

En una operación de escritura se incrementa el índice de escritura.

En una operación de lectura se incrementa el índice de lectura.

En cada operación siempre se incrementa el índice correspondiente.

Cuando los índices coinciden estamos en la condición de COLA VACIA

Cuando el índice de escritura excede el limite estamos en la condición de COLA LLENA

NO SE PUEDE INSPECCIONAR EL CONTENIDO DE UNA COLA, PORQUE SE DESTRUYEN LOS DATOS.

Se creara una función de lectura y una de escritura.

```
void Ingreso(int *Ing,int *Ext,float *Cola)
{
  if(*Ing==MAX)
  {
    printf("\n\nCola llena");
    printf("\nPresione una tecla para continuar");
    getchar();
    return;
  }
  printf("\nIngrese el dato : ");
  scanf("%f",&Cola[*Ing]);
  (*Ing)++;
}
```

```
void Lee(int *Ing,int *Ext,float *Cola)
{
  if(*Ing==*Ext)
  {
    printf("\n\nCola vacia");
    printf("\nPresione una tecla para continuar");
    getchar();
    return;
  }
  printf("\nEl dato es : %f",Cola[*Ext]);
  (*Ext)++;
}
```

Inconvenientes de la cola

Cada vez que se ingresa un dato se achica el tamaño disponible de almacenamiento

Una vez que se llena y se vacia no se puede volver a utilizar

En ese caso se da la condicion de COLA VACIA y COLA LLENA simultaneamente.

La solucion es la COLA CIRCULAR

COLA CIRCULAR

Ambos indices siempre se incrementan.

Cuando un indice supera el limite se produce un SALTO CIRCULAR

Vuelve al inicio

Cuando los dos indices coinciden podemos tener la condicion de COLA LLENA o COLA VACIA

Se soluciona agregando dos banderas

Una que indique cola llena

Otra que indique cola vacia

```c
void Ingre(int *Cola,int **Ing,int *Lle,int *Vac,int *Egr)
{
  if(*Lle)
  {
    printf("\n\nCola llena");
    printf("\n\nPresione una tecla para continuar\n");
    getchar();
    return;
  }
  printf("\n\nIngrese el dato : ");
  scanf("%d",*Ing);
  *Vac=0;
  (*Ing)++;
  if(*Ing==Cola+MAX)
    *Ing=Cola;
  if(*Ing==Egr)
    *Lle=1;
}
```

ing

ing

1

2

egr

3

ing

ing

```c
void Lee(int *Cola,int **Egr,int *Vac,int *Lle,int *Ing)
{
  if(*Vac)
  {
    printf("\n\nCola vacia");
    printf("\n\nPresione una tecla para continuar\n");
    getchar();
    return;
  }
  *Lle=0;
  printf("\n\nEl dato es : %d\n",**Egr);
  (*Egr)++;
  if(*Egr==Cola+MAX)
    *Egr=Cola;
  if(*Egr==Ing)
    *Vac=1;
}
```

PILA - STACK

El Primer dato que se ingresa es el ultimo que se lee.

Se utiliza un puntero o índice.

En una operación de escritura se incrementa el índice de pila.

En una operación de lectura se decrementa el índice de pila.

Cuando el índice de pila apunta al comienzo estamos en la condición de PILA VACIA

Cuando el índice de pila excede el limite estamos en la condición de PILA LLENA

NO SE PUEDE INSPECCIONAR EL CONTENIDO DE UNA PILA, PORQUE SE DESTRUYEN LOS DATOS.

Se creara una función de lectura y una de escritura.

```
void Ingresa(int *Indice,float *Pila)
{
  if(*Indice==MAX)
  {
    printf("\n\nPila llena");
    printf("\n\nPresione una tecla para continuar\n");
    getchar();
    return;
  }
  printf("\nIngrese el dato : ");
  scanf("%f",&Pila[*Indice]);
  (*Indice)++;
}
```

1	← índice
2	← índice
3	← índice
	← índice

```
void Lee(int *Indice,float *Pila)
{
  if(!*Indice)
  {
    printf("\n\nPila vacia");
    printf("\n\nPresione una tecla para continuar\n");
    getchar();
    return;
  }
  (*Indice)--;
  printf("\nEl dato es : %f",Pila[*Indice]);
}
```

1	←	índice
2	←	índice
3	←	índice
	←	índice